WHITEPAPER

Mioty Technology and Implementation Overview

Kiruba Subramani, Roland Gémesi

SILICON LABS

Mioty Introduction

Mioty is a wireless Low Power Wide Area Networking (LPWAN) protocol operating in the 868/915 MHz RF frequency bands that are available worldwide without licensing. Mioty was originally developed by the Fraunhofer IIS [1], and the technology has been adopted as part of the ETSI TS103357 standard [2]. In comparison to the heavily congested 2.4 GHz frequency spectrum, the sub-GHz band offers less interference, longer range, and better propagation through walls and obstacles. As a result, the Mioty protocol, promoted by the Mioty Alliance [3], facilitates achieving several kilometers of wireless transmission range in an energy-efficient manner.

Mioty is suitable for building privately owned massive sensor network deployments, which can scale to hundreds of thousands of battery-operated or energy-harvesting end devices in a star topology. Mioty is specifically designed for end nodes that transmit small amounts of data infrequently, thereby achieving a battery life of up to 15 years. The robustness, scalability, and mobility offered by Mioty make it suitable for a wide range of data acquisition applications where real-time operation is not required, such as remote monitoring, asset tracking, and smart metering.

The following sections provide a technical overview of this wireless standard.

[1] Fraunhofer Institute
[2] ETSI TS 103 357 v1.1. (2018-06) Technical Specification
[3] Mioty Alliance

Mioty: System Architecture and Operation

The system architecture of a Mioty network is shown in Figure 1, and it includes the following components:

- **Endpoints** are battery-powered wireless sensor nodes that periodically transmit field data in the network. Mioty supports two types of endpoints, namely:
  - **Class Z endpoints** support only unidirectional uplink communication, which is suitable for unacknowledged data acquisition. These endpoints must be preconfigured and manually associated with the network by the end user.
  - **Class A endpoints** support bidirectional communication, enabling acknowledged data acquisition, remote reconfiguration, and over-the-air network association.

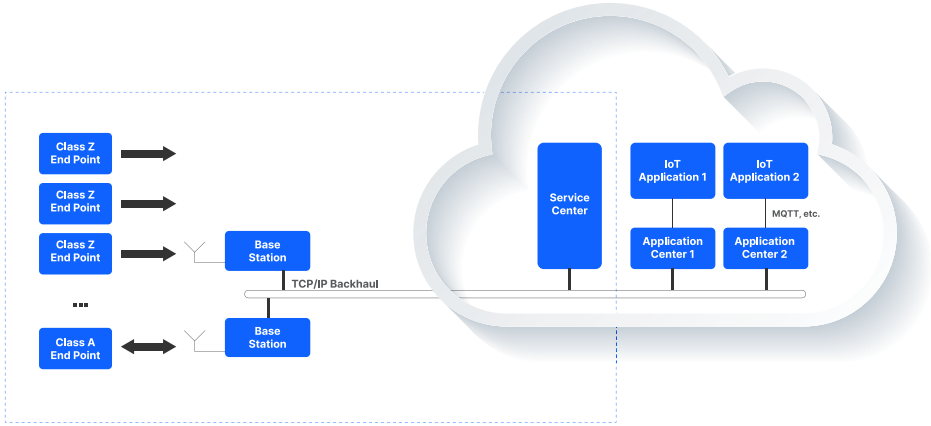


Figure 1. Mioty System Architecture

- **Base stations** receive wireless uplink messages from endpoints and forward them to the backend systems through a high throughput IP-based backhaul, such as Ethernet or Wi-Fi. Mioty deployments typically include multiple base stations to extend coverage and to improve network robustness. Base stations are also responsible for transmitting wireless downlink messages back to class A endpoints.
- **The service center** is a central element of the Mioty backend that is responsible for network management and configuration, such as registering new base stations and endpoints. The service center collects all the uplink data received by the base stations. Since multiple base stations can receive the uplink data transmitted by the same endpoint, the service center is responsible for removing duplicate copies of the received messages. Moreover, it also chooses and directs the appropriate base station to transmit acknowledgments and downlink messages to specific class A endpoints.
- **Application centers** are responsible for decrypting, decoding, and extracting application-specific information from the raw binary data received from the service center. The interpreted information is then forwarded to an IoT application or platform by using IoT protocols such as MQTT.
- **Applications/application platforms** are mostly cloud-based IoT applications whose main functions are data storage, analytics, and visualization.

A Mioty network is formed around one or more privately owned base stations that are registered with the service center, which is the central network management system that is either running on the cloud or at a privately owned infrastructure.

To join a Mioty network, endpoints need to be registered with the service center through the attachment procedure. While unidirectional (class Z) endpoints are manually attached to the network, bidirectional (class A) endpoints can initiate an over-the-air attachment procedure.

Some of the features differentiating Mioty from other protocols are described below:

- **Telegram Splitting:** The core feature of Mioty is Telegram Splitting Multiple Access (TSMA), where the application data is divided into multiple sub-packets and transmitted as short radio bursts. These transmissions are carried over at different time instances and over different frequency channels by following a hopping sequence, making the communication resilient to interference.
- **Coded PHY:** Another core feature of Mioty is the use of Forward Error Correction (FEC) coding in the physical layer. Mioty uses 1/3 rate convolutional code, which allows the communication to lose up to 50% of the transmitted packets and still recover the original message.
- **Short Addresses:** Although each endpoint has a 64-bit long unique address (EUID64), a 16-bit non-unique short address is used to minimize communication overhead. At the receiver, the transmitting endpoint is determined by matching the message signature with a list of potential transmitters.
- **Random Channel Access:** Devices in a Mioty network do not follow a coordinated channel access scheme. Instead, individual endpoints can initiate uplink communication anytime, and the protocol tolerates interference and packet loss by relying on telegram splitting and coded PHY as described above. Downlink communication for class A devices, on the other hand, starts at a pre-defined time after an uplink communication.

Supported by the above techniques, Mioty networks can achieve up to 15 km of range in line-of-sight communication in a rural environment and up to 5 km of range in an urban environment. Although the protocol is optimized for transmitting 10 bytes of application data to the base station, it is capable of supporting up to 245 bytes in the uplink and up to 250 bytes in the downlink communication. Moreover, since the protocol communicates through short radio bursts, it causes only subtle battery voltage drops, thereby increasing the overall usable battery capacity. This allows Mioty devices to achieve up to 15 years of battery life or makes them suitable for energy harvesting-based applications. Finally, towards scalability, a Mioty base station can support up to 110,000 endpoints and handle up to 3.5 million messages per day. [4]

[4] Mioty – The all-around talent for industrial IoT applications handles 3.5 million messages per day

Mioty Protocol Stack

The protocol stack of Mioty is shown in Figure 2 and explained below.

Wireless Networking Stack (LPWAN)	
Application / Application Support	IoT Application running on Cloud (Customizable via Application Center)
Transport Layer	TCP/IP Backhaul (Not defined in Mioty)
Network Layer	Logical Link Control (LLC), Base Station Service Center Interface (BSSCI), Service Center Application Center Interface (SCACI)
Data Link Layer	Uplink: Random Channel Access, Downlink: Pull-based Time Scheduled, Address Resolution
Physical Layer	ETSI 103357 TS-ULP, 898/915 MHz, (G)MSK, Telegram Splitting Ultra Narrow Band (TS-UNB), Channel hopping, Forward Error Correction (FEC), 2380 symbols/s

Figure 2. Mioty Protocol Stack

The physical layer of a Mioty device is according to the Telegram Splitting Ultra Narrow Band (TS-UNB) protocol family as defined in ETSI 103357 [2]. The protocol operates in the 898/915 MHz frequency bands in the EU/US regions, and the available bands are further divided into 25 sub-channels of 3 kHz bandwidth each.

The protocol uses Gaussian Minimum Shift Keying (GMSK) modulation, which allows it to use non-linear but efficient power amplifiers in the RF front end. It also encodes the transmitted messages using a 1/3 rate convolutional code-based Forward Error Correction (FEC), which allows it to recover messages even when up to 50% of the radio bursts are affected by collisions and RF interference.

As shown in Figure 3, the uplink transmission of an endpoint starts with a core frame, which can contain up to 10 bytes of application data and is optionally followed by an extension frame for transmitting additional data. The core frame consists of 24 radio bursts that are transmitted over 24 different carrier frequencies based on a random hopping sequence. The radio bursts are 15.14 ms long and spaced 160 ms apart, resulting in a total transmission time of approximately 3.7 s for a core frame. When transmitting the extension frame, the endpoint can transmit up to 235 radio bursts, each containing a single byte of application data. The radio bursts of the extension frame are distributed over all the 25 carrier frequencies with similar timing to the core frame.

In the case of a class A device, a base station can perform a downlink communication back to the endpoint. The timing between the uplink and downlink communication is specified as 16384 symbols, which equals 6.9 s. The downlink core frame is 9 radio bursts long and is used only to acknowledge the uplink data without transmitting any application data. The core frame can be optionally followed by multiple extension frames, each consisting of 18 radio bursts. Each extension frame can transmit up to 24 bytes of application data, and the standard limits the overall downlink communication to a maximum of 250 bytes.

[2] ETSI TS 103 357 v1.1. (2018-06) Technical Specification

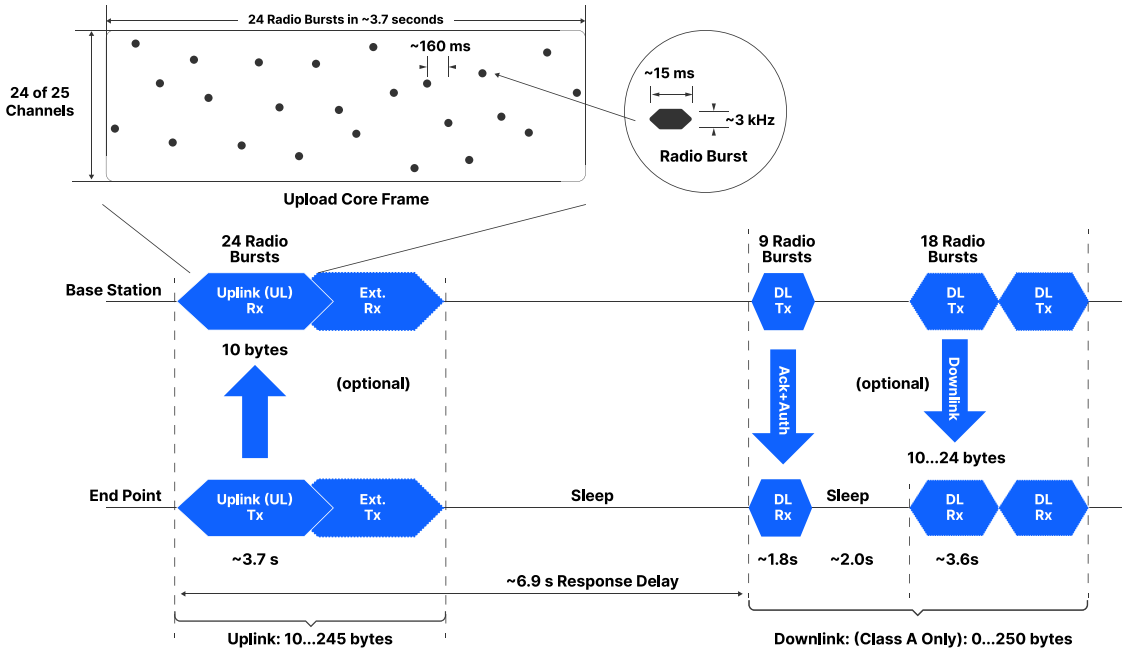


Figure 3. Telegram Splitting in Mioty Transmission

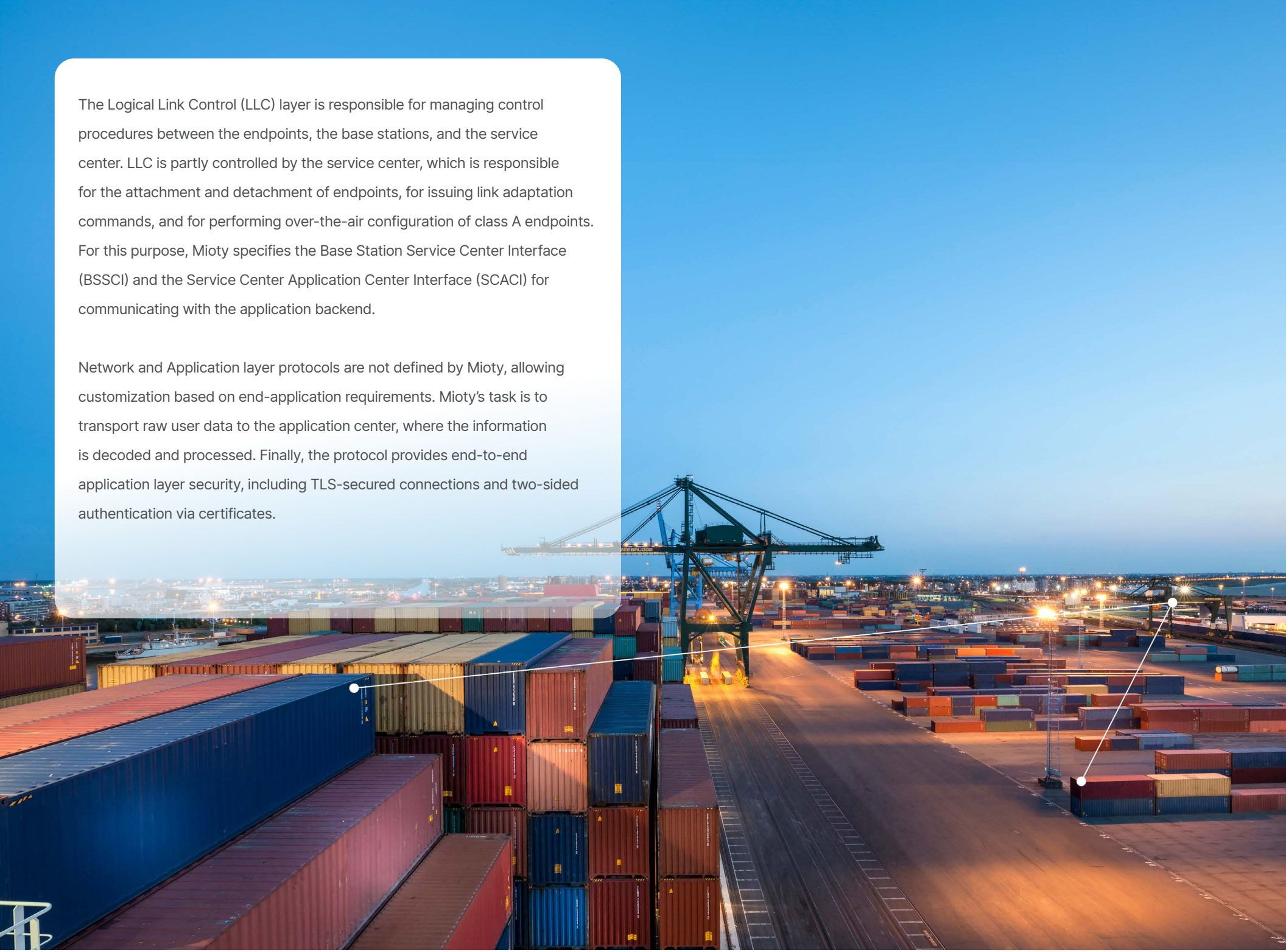
The MAC layer [5] of Mioty allows endpoints to initiate uplink transmission at any time without channel assessment. This reduces the complexity of the MAC layer implementation and allows immediate access to the communication channel at the cost of collisions. However, the effect of such collisions is minimized by the various PHY techniques described earlier. The MAC layer is responsible for storing uplink message timestamps and controlling the downlink time windows so that the downlink messages are issued at the right time.

The MAC layer is also responsible for address resolution. Each endpoint has a lifetime unique EUI64 address, but the uplink message includes only a 16-bit non-unique short address, which is assigned during the attachment procedure. The Cipher-based Message Authentication Code (CMAC), which is derived using the EUI64 long address, plays a critical role in the address resolution at the receiving base station. Therefore, when a message is received at the base station, the MAC layer uses the CMAC verification procedure to identify the actual transmitting endpoint from a list of potential candidates. Additionally, by decoding the CMAC, the MAC layer verifies the authenticity and integrity of the message. The confidentiality of the transmitted messages is protected by using AES128-based encryption mechanisms. The MAC layer is accountable for decrypting messages that are protected by this network-level security so that data can then be interpreted and processed by the higher protocol layers.

[5] Mioty– Mac and Higher Layers

The Logical Link Control (LLC) layer is responsible for managing control procedures between the endpoints, the base stations, and the service center. LLC is partly controlled by the service center, which is responsible for the attachment and detachment of endpoints, for issuing link adaptation commands, and for performing over-the-air configuration of class A endpoints. For this purpose, Mioty specifies the Base Station Service Center Interface (BSSCI) and the Service Center Application Center Interface (SCACI) for communicating with the application backend.

Network and Application layer protocols are not defined by Mioty, allowing customization based on end-application requirements. Mioty's task is to transport raw user data to the application center, where the information is decoded and processed. Finally, the protocol provides end-to-end application layer security, including TLS-secured connections and two-sided authentication via certificates.



Implementing Mioty on a Silicon Labs SoC

Silicon Labs offers high-performance sub-GHz wireless SoCs, such as the EFR32FG23 [6], which is an ideal candidate for a Mioty implementation [7]. The SoC is optimized for the 868/915 MHz wireless bands, where it offers best-in-class receiver sensitivity. It includes up to 512 kB flash storage and 64 kB RAM capacity, which enables executing the Mioty protocol stack along with the application code on a single SoC, resulting in an integrated solution.

The EFR32FG23 is specially optimized for low-power applications. In active mode, operating at 78 MHz clock speed, the SoC consumes 2.1 mA supply current. In ultra-low power mode, on the other hand, the SoC consumes only 1.2 µA sleep current while retaining 16 kB of RAM content. For Mioty endpoint implementation, maximum transmit power and the corresponding current consumption are critical parameters in determining the overall battery life. To this end, EFR32FG23 supports +20 dBm transmit power and consumes 25 mA current during transmissions. This allows typical Mioty end-point implementations using EFR32FG23 to achieve 15 years of battery life, as can be seen in the table below.

Table 1 considers various application data sizes and shows the corresponding on-air time, total transmission time, energy consumption per message, and the transmission interval to achieve 15 years of battery life, assuming a 2200 mAh battery. For instance, when transmitting 10-byte application data, the on-air time of the message is 363 ms, the transmission duration is 3.7 s, the charge needed for transmission is 2.5 mAh, and the device can transmit a message every 12 minutes to achieve a 15-year battery lifetime. Estimates for application data sizes of 50 bytes and 200 bytes are also provided in the table.

Application Data Size	Message On-air Time	Total Message Time	Charge per Message	Transmission Interval to achieve 15 Years of Battery Life
10 bytes	363 ms	3.7 s	2.5 µAh	5 msg / hour
50 bytes	969 ms	10.1 s	6.7 µAh	2 msg / hour
200 bytes	3240 ms	34.1 s	22.5 µAh	1 msg / 2 hour

Table 1. Mioty Uplink Transmissions
(I_tx=25mA; I_sleep=1.2uA; Battery capacity=2200mAh; Battery self-discharge=20%)

On the software side, Fraunhofer IIS provides an implementation of the Mioty wireless stack. The provided Mioty core library implements all the necessary platform-independent radio protocol functions. Moreover, Fraunhofer IIS also provides an EFR32FG23-specific hardware abstraction layer for the configuration and control of the Silicon Labs hardware. To easily integrate the stack with the application code, software API and an AT commands library are also provided. The Mioty software stack shall be licensed via the Sisvel [8] platform, which offers various pricing options for building class Z and class A endpoints and gateways.

To decrease development efforts, Swissphone [9] offers an off-the-shelf Mioty module that uses the EFR32FG23 SoC. The module has a tiny LLC form factor of 18×14.2×2.8 mm and is capable of operating in both the EU and US regions. The module enables building class Z or class A endpoints without requiring in-depth knowledge of the Mioty wireless technology.

[6] EFR32FG23 Wireless SoC Family Data Sheet
[7] FRAUNHOFER IIS: Cooperation with Silicon Labs
[8] Sisvel Mioty Licensing Platform
[9] Swissphone Mioty Module

Summary

Mioty is a wireless Low Power Wide Area Networking (LPWAN) protocol operating in the sub-GHz RF band. It is based on the Telegram Splitting Ultra Narrow Band (TS-UNB) standard, specifically designed for low data rate and infrequent communications, thereby offering kilometers of range and up to 15 years of battery life. This makes Mioty an ideal wireless technology for long-range data acquisition systems, such as remote monitoring. Silicon Labs offers sub-GHz wireless SoCs, such as the EFR32FG23, which features best-in-class sensitivity and energy consumption figures, making it an ideal candidate for Mioty implementation. In addition, the availability of the software stack and the off-the-shelf module solution simplifies device implementation and results in fast development cycles.

SILICON LABS + mioty alliance